

# Computer vision Seminar

수원대학교  
Data Network Analysis 

---

데이터과학부  
정인호

2023.02.02

# Week3) ImageNet Challenge 2014

---

1. Week 2 과제 리뷰
2. 파이썬 이미지 데이터 다루기
3. ImageNet Challenge 2014
4. Week 3 과제 설명

# Week3) ImageNet Challenge 2014

---

## 1. Week 2 과제 리뷰

- 과제 피드백 – 가독성이 높은 코드

# 1. Week 2 과제 리뷰

과제 피드백 – 가독성이 높은 코드

```
1 import glob
2 '''glob 함수를 이용하면, 해당 경로에 있는 파일들의 경로를 전부 리스트로 얻어올 수 있습니다. glob 라이브러리를 이용하여 이번 과제를 해결해주세요.'''
3 dolphin_img_list = glob.glob('/content/drive/MyDrive/CV_seminar_project/original/dolphin/*')
4 shark_img_list = glob.glob('/content/drive/MyDrive/CV_seminar_project/original/shark/*')
5 whale_img_list = glob.glob('/content/drive/MyDrive/CV_seminar_project/original/whale/*')
6
7 dic = {'dolphin':dolphin_img_list, 'shark': shark_img_list, 'whale': whale_img_list}
8 for key in dic.keys():
9     print(f'{key}이미지가 ',len(dic[key]), '개 있습니다.')
10 print('-----')
11
12 length_list = []
13 for key in dic.keys():
14     print(f'{key}이미지는 trian, valid, test셋에 대해 ',int(len(dic[key])*0.7), int(len(dic[key])*0.2), int(len(dic[key])*0.1), '개씩 배정해주세요.')
15     length_list.append([int(len(dic[key])*0.7), int(len(dic[key])*0.2), int(len(dic[key])*0.1)])

dolphin이미지가 373 개 있습니다.
shark이미지가 488 개 있습니다.
whale이미지가 451 개 있습니다.

-----

dolphin이미지는 trian, valid, test셋에 대해 261 74 37 개씩 배정해주세요.
shark이미지는 trian, valid, test셋에 대해 341 97 48 개씩 배정해주세요.
whale이미지는 trian, valid, test셋에 대해 315 90 45 개씩 배정해주세요.
```

# 1. Week 2 과제 리뷰

과제 피드백 – 가독성이 높은 코드

```
1 import os
2 import glob
3 import cv2
4 import time
5
6 class Make_dataset_dir():
7     def __init__(self, root_dir):
8         '''클래스 내의 인자 생성 함수입니다. 아래의 인자들을 이용하여, 금주의 과제를 해결해주세요.'''
9         self.root_path = root_dir+'/' if root_dir[-1] != '/' else root_dir # 현재 진행할 프로젝트 -> root path는 /content/drive/MyDrive/CV_seminar_project/ 가 되어야합니다.
10        self.img_path_list = root_dir+'original' # 전달한 이미지들의 상위 경로
11        self.trainset_path = root_dir+'train/'
12        self.validset_path = root_dir+'valid/'
13        self.testset_path = root_dir+'test/'
14        self.class_list = ['dolphin', 'shark', 'whale']
15
16    def mk_dir(self):
17        '''train, valid, test 폴더를 만들고, 내부에는 클래스 별 폴더를 추가로 만들어 주세요.'''
18
19    def move_img(self):
20        '''mk_dir에서 만든 폴더들에 각 클래스에 맞는 이미지를 배당해주세요. train, valid, test에 각각 7: 2: 1'''
21
22    def run(self):
23        '''run 함수를 돌리면, 모든 작업이 완료될 수 있게끔 mk_dir 함수와 move_img 함수를 생성해주시면 됩니다.'''
24        start = time.time()
25        self.mk_dir()
26        self.move_img()
27        print('총 소요시간: ', time.time()-start)
28
29    def checking_dirs(self):
30        '''데이터가 잘 스플릿 되었는지 확인하는 함수입니다. train, valid, test 디렉토리에 데이터가 잘 들어가 있는지 개수를 확인하는 함수를 생성해주시면 됩니다.'''
```

# Week3) ImageNet Challenge 2014

---

## 2. 파이썬 이미지 데이터 다루기

- 라이브러리 다루기 (실습)
- Github 코드 업로드 (실습)

## 2. 파이썬 이미지 데이터 다루기

이미지 데이터 핸들링

[과제 풀이 링크](#)

**Week2 과제 파일을 열어 실습을 진행합니다.**

**이미지 데이터를 다루는 3가지 라이브러리**

# Week3) ImageNet Challenge 2014

---

## 3. VGG Net, GoogLeNet

- VGG Net
- GoogLeNet



## CNN의 주요 용어 정리

- Convolution(합성곱)
- 채널(Channel)
- 필터(Filter)
- 스트라이드(Strid)
- 피쳐 맵(Feature Map)
- 액티베이션 맵(Activation Map)
- 패딩(Padding)
- 풀링(Pooling) 레이어

# 3. ImageNet Challenge 2014

VGG Net & GoogLeNet

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners  
cs231n 中 ...

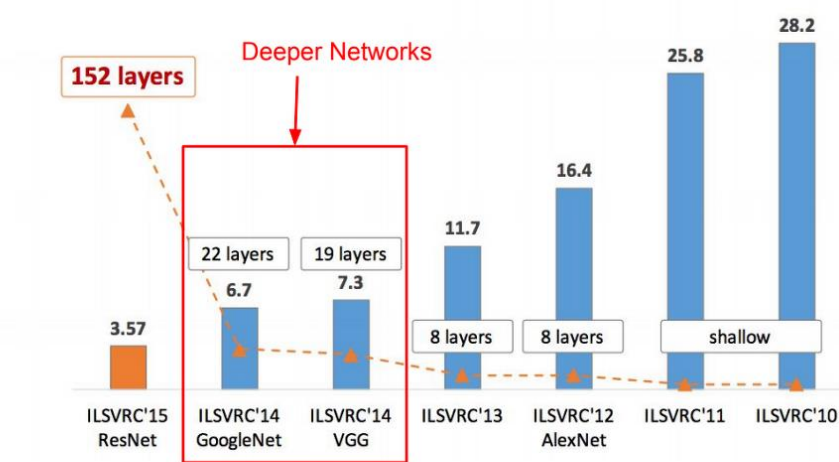


Figure copyright Kaiming He, 2016. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 9 - 25

May 2, 2017

ImageNet Challenge 2014에서 전년도 대비 큰 성능 차이를 보여준 2개의 모델

# 3. ImageNet Challenge 2014

VGG Net

[Vggnet 논문](#)

**논문 초록을 읽어보자.**

# 3. ImageNet Challenge 2014

VGG Net

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 LRN	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

ImageNet Challenge 2014에서 전년도 대비 큰 성능 차이를 보여준 2개의 모델

# 3. ImageNet Challenge 2014

VGG Net

## 2.3 Discussion

vggnet의 configuration은 이제껏 우승했던 모델(AlexNet)과는 다르다고합니다.

첫 Conv layer를 비교적 큰 사이즈인 7x7, stride는 2를 쓴 AlexNet과는 달리 vggnet 저자들은 전체 모델에 stride는 1이며, 아주 작은 3x3필터를 쓴게 포인트라고 합니다.

### Case Study: VGGNet

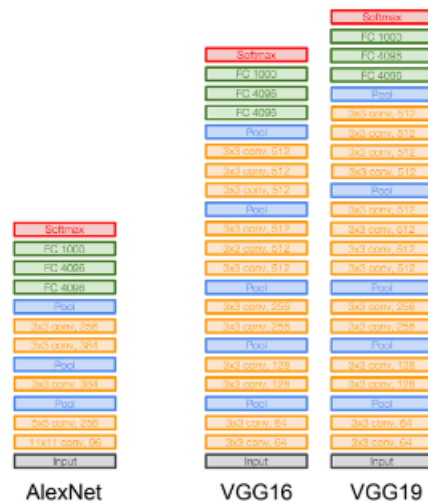
[Simonyan and Zisserman, 2014]

Q: Why use smaller filters? (3x3 conv)

Stack of three 3x3 conv (stride 1) layers has same **effective receptive field** as one 7x7 conv layer

But deeper, more non-linearities

And fewer parameters:  $3 * (3^2 C^2)$  vs.  $7^2 C^2$  for C channels per layer



풀링없이 3x3을 2번 쓰면 5x5 필터를 1번 쓴 것과 같고, 7x7은 3x3 필터를 3번 쓴 효과와 같다고 합니다.

그래서 7x7대신에 3x3을 써서 얻을 수 이점이 있을까요?

**1. 비선형 함수인 relu를 3번이나 쓸 수 있습니다.**

(비선형 함수를 3번씩이나 쓰면 모델이 좀 더 깊어져서, 데이터에 좀 더 유연하게 대응할 수 있겠죠?)

**2. 학습 파라미터의 수가 감소**

3x3 filter를 3번 쓸 때의 파라미터 수 =  $n\_layer \times (H \times W \times C) = 3(3 \times 3 \times C) = 27C^2$

7x7 filter를 1번 쓸 때의 파라미터 수 =  $n\_layer \times (H \times W \times C) = 1(7 \times 7 \times C) = 49C^2$

(채널은 입력 데이터의 채널과 필터의 채널을 모두 고려합니다.)

81%나 파라미터 수를 감소시킬 수 있습니다.

**1x1 Conv layer**는 어떤 영향을 주는 receptive fields Conv layer가 아니지만, 비선형성을 가해줄 수 있는 방식중에 하나입니다. 또 같은 dimension의 공간으로 선형 프로젝션을 해주는 용도로도 이용할 수 있습니다.

(위에서 제가 말했던 것)

vggnet논문에서 당시 우승했던 GoogLeNet을 언급합니다.

구글넷은 우리(vggnet)와 독립적으로 연구가 됐지만 Conv layer를 깊게 쌓았다는 점이 우리와 유사했고, 그로인해 좋은 성능을 얻었습니다. 하지만 vggnet에 비해 훨씬 complex합니다. 첫번째 layer부터 연산량을 줄이기 위해 공간적인 보존 (아마 filter의 사이즈를 의미할 것)을 공격적으로 줄였습니다.

# 3. ImageNet Challenge 2014

VGG Net

**VGG Net 정리하기**

# 3. ImageNet Challenge 2014

GoogLeNet

GoogLeNet 논문

**논문 초록을 읽어보자.**

구글넷 블로그 포스팅: <https://inhovation97.tistory.com/45>

# 3. ImageNet Challenge 2014

GoogLeNet

## < 2. 저자들의 실험 초점 >

저자들이 연구한 내내 신경썼던 방향성이 있을 겁니다.

**2. related work & 3. Motivation and High Level Considerations** 부분의 핵심 내용을 설명합니다.

(2. related work의 1x1 Conv와 다양한 사이즈의 Conv의 내용이 대략적으로 나오지만 아키텍처 부분에서 설명합니다.)

구글팀도 모델을 깊게 쌓기 위해서 노력을 하는데, depth와 width를 언급합니다. 아키텍처의 그림을 보면 아시겠지만, 여러 사이즈의 Conv를 쓰기 때문에 **width**라는 단어가 나옵니다.

### 모델을 깊게 쌓기위한 문제점

1. 모델을 깊게 쌓으면, 당연히 오버피팅의 문제가 존재합니다.

또한 모델이 너무 깊으면 모델의 일반화도 힘들어져서 학습셋이 적은 경우는 이미지의 디테일한 부분을 잡기가 더 더욱 힘들어집니다.

2. 네트워크의 사이즈가 깊어질수록 연산량은 훨씬 가파른 기율기로 증가합니다.

3x3의 Conv를 2개 쌓는다고 예를들면,

HxWxC가 3x3을 거쳐 HxWxC1로 나옵니다. 이걸 다시 Conv를 거치면 HxWxC2가 되는데, C1채널로 나온 피쳐맵을 또 다시 C2로 바꿔주니까 layer를 한개 더 쌓았을 뿐인데 계산량은 제곱으로 늘어나는거죠.

**네트워크를 깊게 쌓되, 효율적으로 깊게 쌓아야 한다는겁니다.**



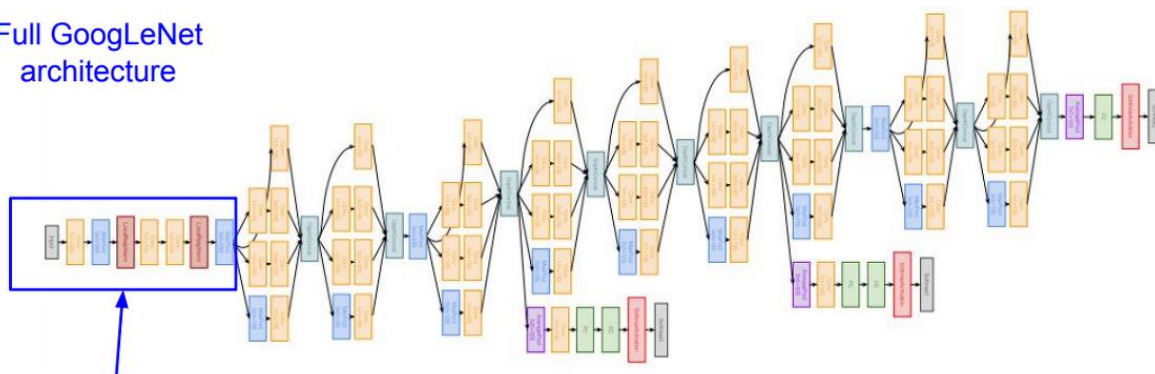
# 3. ImageNet Challenge 2014

GoogLeNet

## Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet  
architecture



Stem Network:  
Conv-Pool-  
2x Conv-Pool

처음 부분은 AlexNet의 구조를 그대로 가져옵니다.

vggnet과 큰 차이점은 vggnet은 처음에 큰 사이즈의 filter로 합성곱을 하는 것에 부정적이었습니다.

vggnet의 configuration은 이제껏 우승했던 모델(AlexNet)과는 다르다고합니다.

첫 Conv layer를 비교적 큰 사이즈인 7x7, stride는 2를 쓴 AlexNet과는 달리 vggnet 저자들은 전체 모델에 stride는 1이며, 아주 작은 3x3필터를 쓴게 포인트라고 합니다.

<https://inhovation97.tistory.com/44>

vggnet은 모델의 효율성을 위해 작은 필터만을 고집했지만, 구글넷 저자들은 관점이 달랐습니다.

input에 가까운 layer들은 굉장히 적은 범위에 correlated unit들이 분포하고 집중돼있다고 합니다.

이걸 제 생각대로 해석해보자면, 이미지 사이즈는 input일 때에 가장 사이즈가 큼니다.

그러다보니, 필터들이 그것을 전부 훑으면, 낭비되는 값들이 당연히 많을 겁니다. 그래서 AlexNet처럼 일단은 resolution을 큰 filter로 크게 줄여버리고나서 본격적인 인셉션 모듈로 들어가자는 의도였던 것 같습니다.

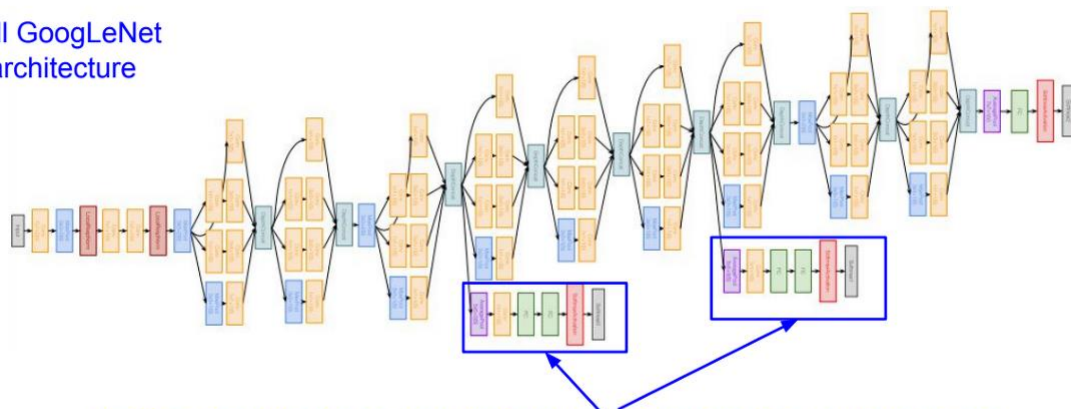
# 3. ImageNet Challenge 2014

GoogLeNet

## Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet  
architecture



Auxiliary classification outputs to inject additional gradient at lower layers  
(AvgPool-1x1Conv-FC-FC-Softmax)

중간 중간에 손잡이 모양의 분류기가 있습니다.

저자들은 **Auxiliary classifier**라고 하는데, 이것을 추가함으로써 backward 과정에서 깊은 네트워크의 기울기 소실 문제를 방지할 수 있습니다.

이 보조 분류기에도 똑같이 softmax 1000 class output을 설정하는데, train시에는 Auxiliary classifier에서의 loss가 total loss에 더해지는데 0.3의 가중치를 곱해 영향력을 낮춰서 더해줍니다. 그렇기 때문에 저자들은 regularization의 효과도 기대해 볼 수 있다고 합니다. 약간의 앙상블(?) 효과도 기대해 볼 수 있는거죠.

추론 시에는 쓰지 않는다고합니다.

학습때만 이용합니다.

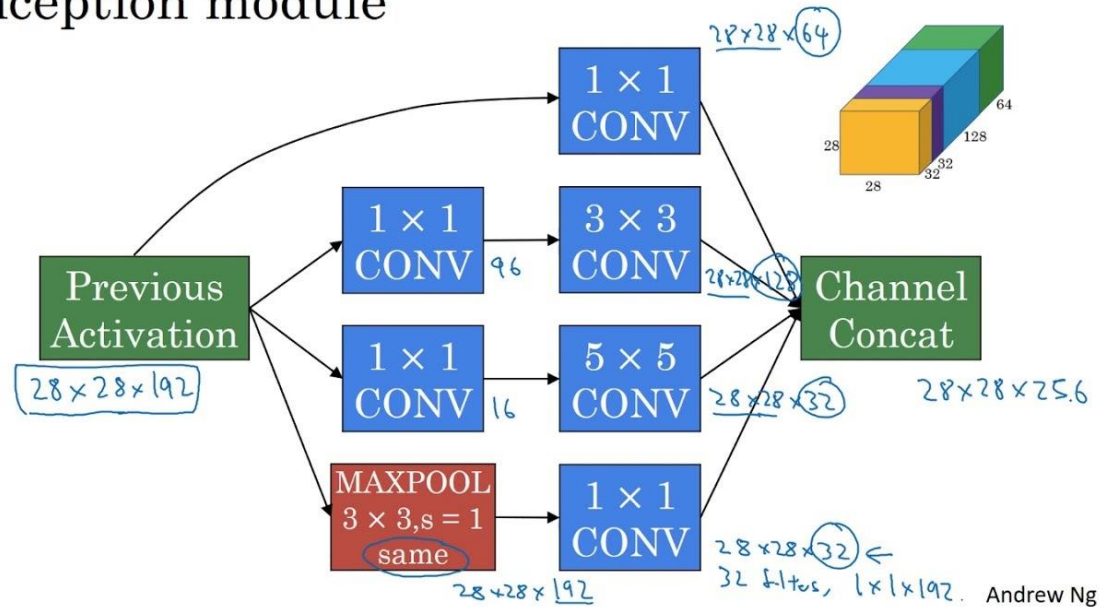
보조 분류기, 최종 분류기를 보면 **Global Average Pooling**이 쓰입니다.

# 3. ImageNet Challenge 2014

GoogLeNet

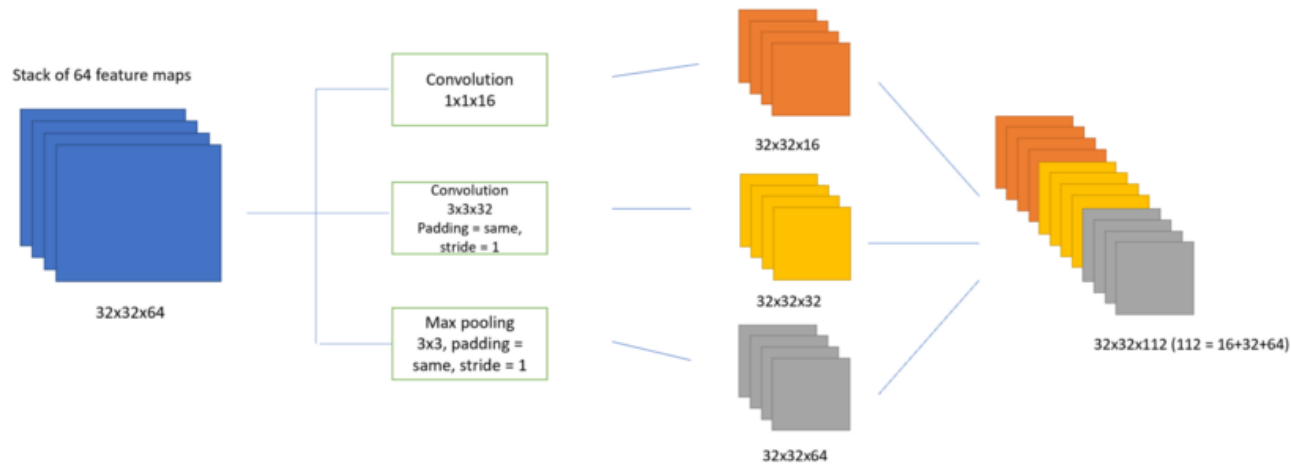
Inception module

CNN width를 넓힌 GoogLeNet



# 3. ImageNet Challenge 2014

GoogLeNet



<https://valentinaalto.medium.com/understanding-the-inception-module-in-googlenet-2e1b7c406106>

# 3. ImageNet Challenge 2014

GoogLeNet

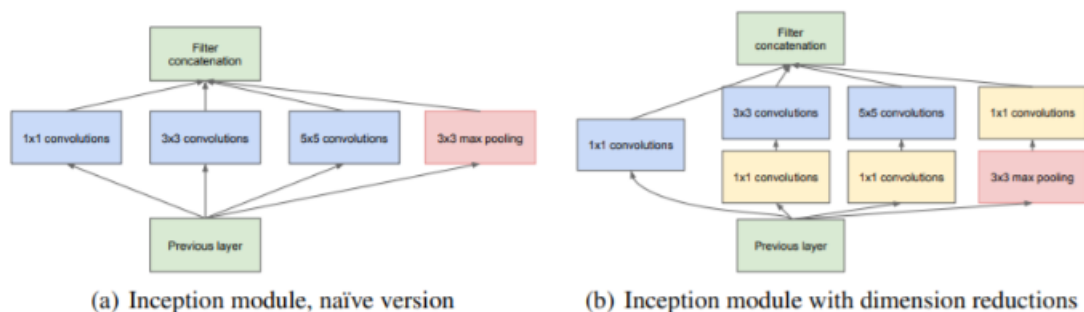


Figure 2: Inception module

그렇게 인셉션 모듈을 만들고, naïve version이 아닌 b를 선택하게 됩니다.

5x5 filter를 사용하면, 연산량이 너무 커지게 됩니다. 그래서 오른쪽과 같이 1x1 Conv로 channel reduction을 해줍니다.

1x1 filter가 vggnet에서도 그렇고 굉장히 중요한 역할을 한다는 것을 알 수 있습니다.

위 방식은 Cross Channel Pooling방식입니다.

maxpooling을 채널별 픽셀로 해줌으로써 같은 사이즈의 feature map을 원하는 사이즈로 down시킬 수 있는 기법입니다.

1x1 Conv는 학습할 수 있는 파라미터가 있습니다. 즉 down size도 좀 더 의미있게 할 수 있고, relu 함수도 거치기 때문에 비선형성도 한번 더 가해줄 수 있습니다. 여러 필터 사이즈로 나온 feature map들을 채널 수를 줄일 때 또 한 번 그것에 대해 고려해줄 수 있는 비선형성을 고려해줄 수 있으니 아주 좋은 방식인 거죠.

3x3 max pooling은 좀 더 다양한 resolution을 위해 추가해준 것이라고 합니다.

3x3 맥스 풀링을 할 때, 나와있지는 않지만 아마 사이즈 다운을 방지하기 위해 stride를 겹치게 할 겁니다.

그럴 경우 계속해서 특징이 강한 의미있는 픽셀의 값이 여러 번 겹치기 때문에 또 그런 부분의 장점이 있을 겁니다.

filter들의 output channel 같은 하이퍼 파라미터들은 layer depth마다 다르게 설정해줍니다.

(개인적으로는 이미 다른 발전된 모델은 보고나서 느낀 점은 이때의 구글넷이 너무 헤비해서 쓸모없는 정보도 많이 포함돼있지 않나 싶습니다. 이렇게 까지도 비교적 간단한 vggnet과 크게 차이가 나지 않았고, backbone으로도 vggnet이 더 많이 쓰입니다.)

# 3. ImageNet Challenge 2014

GoogLeNet

**GoogLeNet 정리하기**

# Week3) ImageNet Challenge 2014

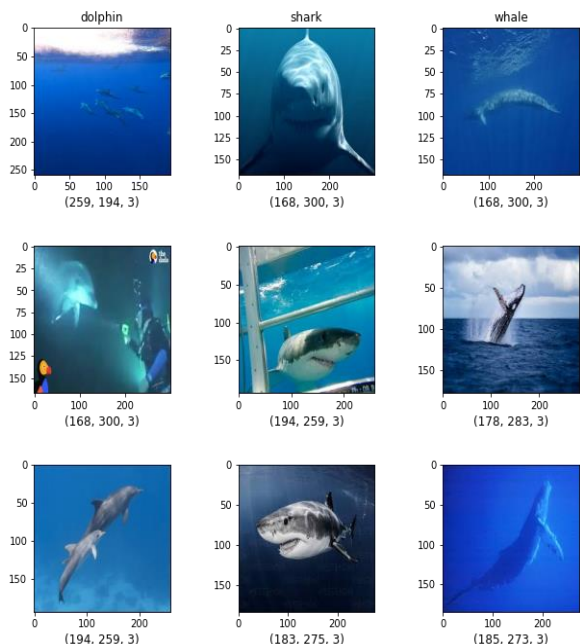
---

## 4. Week 3 과제 설명

# 4. Week 3 과제 설명

## week 3 과제 설명

### 1번 과제 결과물



1번 결과물을 위해서는  
반드시 1개의 for문 만을 이용하여  
시각화 해올 것.  
( 2중 for문도 괜찮음 )

For 문을 이용하여, 이미지 시각화 해보기.  
(본인의 깃헙에 올려주세요.)

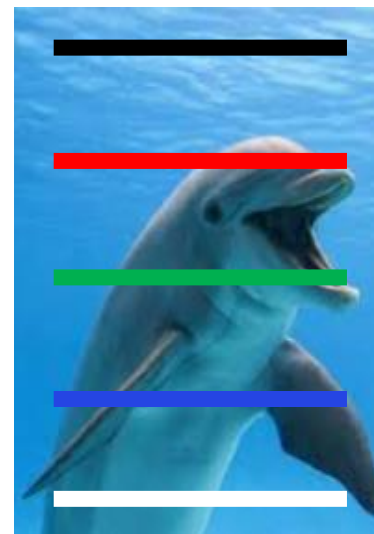
양쪽의 두 결과물을 만들어 오면 됩니다.

1. 반드시 아래의 조건을 지켜주세요 !

2. 제발 코랩 공유 가능한 링크를 저에게 제출해주세요 !  
(혹은 깃헙에 올린 뒤 깃헙 코드 링크를 주시면 됩니다.)

3. 기한을 지켜주세요. (다음 주 수요일 20시 까지)

### 2번 과제 결과물



1. 아무 이미지나 불러와도 된다.

2. cv2 라이브러리에서의 line을  
만드는 함수를 이용하면 안되며,  
반드시 imread를 한 뒤, 직접 픽  
셀 값을 변경하여 위 이미지를 시  
각화 하도록 한다.